CS 134: Networks Lecture 12 outline, Monday, 3/20

Prof. Yaron Singer Spring 2017

1 Overview

In the previous lectures we discussed models of influence and contagion in networks. We proved various structural properties about the parameters of the models and network structure that ensured that an influence process infects the entire population. In the next two lectures we will be concerned with the algorithmic question of how to select the seed set of initial adopters so that influence in the network is maximized.

Question: *How do we select influencers in a network?*

The main result we will discuss today is due to Kempe, Kleinberg and Tardos [2].

2 Influence Maximization in the Independent Cascade Model

Influence maximization is the task of selecting k individuals in the network s.t. the expected influence in the network will be maximized after t steps. We can phrase the influence maximization question in the independent cascade model for the case where $t = |V|$, when the process terminates (all the results that we will discuss in this lecture can be easily modified for an arbitrary selection of t).

3 Submodular Functions

We will later show that in the independent cascade model the expected number of individuals who are influenced is a *monotone submodular* function.

Definition. *A function $f : 2^N \to \mathbb{R}$, is* ***monotone*** *if $S \subseteq T \implies f(S) \leq f(T)$.*

The emphmarginal contribution of an element is the value an element adds to an existing set.

Definition. *Given a function $f : 2^N \to \mathbb{R}$, the* ***marginal contribution*** *of an element $e \in N$ to $S \subseteq N$ is $f_S(e) = f(S \cup e) - f(S)$.*

A submodular function is simply a function that has a *diminishing returns property*: adding more elements to a set means that the the marginal contribution of an element decreases.

Definition. *A function $f : 2^N \to \mathbb{R}$ is* ***submodular*** *if $\forall S \subseteq T \subseteq N$ and $a \notin T$ we have:*

$$f_S(a) \geq f_T(a)$$

Algorithm 1 Greedy Algorithm

```
1: Set $S = \emptyset$
2: while $|S| \leq k$ do
3:     $S \leftarrow S \cup \text{argmax}_{a \in N} f_S(a)$
4: end while
5: return $S$
```

3.1 Properties of submodular functions

First we define *subadditive functions*:

Definition. *A function $f : 2^N \rightarrow \mathbb{R}$ is* ***subadditive*** *if for any $S, T \subseteq N$ we have that:*

$$f(S \cup T) \leq f(S) + f(T).$$

Properties of submodular functions. The following properties will be useful in our analysis:

- If f, g are monotone submodular functions, and $\alpha, \beta > 0$ then:

$$h(S) = \alpha f(S) + \beta g(S)$$

  is a monotone submodular function as well;

- a function $f : 2^N \rightarrow \mathbb{R}$ is submodular if and only if for every $S \subseteq N$ the marginal contribution function $f_S(T) = f(S \cup T) - f(T)$ is subadditive;

Analysis of the greedy algorithm. The above algorithm is simple and intuitive: at every step, simply add the element that has the largest marginal contribution to the set of elements selected. We will now show that the algorithm has a good approximation ratio. In particular, we will show:

Theorem 1 ([3])**.** *For any non-negative monotone submodular function $f : 2^N \rightarrow \mathbb{R}_+$, define $\texttt{OPT} = \max_{|T| \leq k} f(T)$. Then, the greedy algorithm returns a solution S s.t. $f(S) \geq (1 - 1/e)\texttt{OPT}$.*

4 Approximation Algorithms

Should we be happy with the fact that we got a $1 - 1/e$ approximation to our problem? Unfortunately, even very simple cases of independent cascade are NP-hard to optimize. Since the problem is NP-hard, we turned to an approximation algorithm.

Definition. *Let Π be a maximization problem of size n. An algorithm is a $g(n)$-approximation if it reruns a valid solution for Π whose value is at least a $g(n) \leq 1$ fraction of the optimal solution to Π.*

For NP-hard problems, when possible, we seek *constant factor* approximation algorithms that run in polynomial-time. A constant factor approximation implies that the approximation guarantee remains constant, irrespective of the input size. So, while the input size to the problem may grow, the ratio between the optimal solution and the solution returned by the algorithm will not change.

References

[1] U. Feige A threshold of ln n for approximating set cover. *Journal of the ACM*, 45, 634 652.

[2] D. Kempe, J. Kleinberg, E. Tardos. Maximizing the Spread of Influence through a Social Network. *ACM SIGKDD*, 2003. http://www-bcf.usc.edu/~dkempe/publications/spread.pdf

[3] G.L. Nemhauser, L.A. Wolsey, M.L. Fisher. An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming*, 14 (1978), 265294.